アプリケーションを復元する
Trident

NetApp
March 05, 2026

This PDF was generated from https://docs.netapp.com/ja-jp/trident-2506/trident-protect/trident-protect-restore-apps.html on March 05, 2026. Always check docs.netapp.com for the latest.

目次

アプリケーションを復元する

Trident Protectを使用してアプリケーションを復元する

Trident Protect を使用して、スナップショットまたはバックアップからアプリケーションを復元できます。アプリケーションを同じクラスターに復元する場合、既存のスナップショットからの復元の方が高速になります。

- アプリケーションを復元すると、アプリケーションに対して設定されているすべての実行フックもアプリとともに復元されます。復元後の実行フックが存在する場合、復元操作の一部として自動的に実行されます。
- qtree ボリュームでは、バックアップから別の名前空間または元の名前空間への復元がサポートされています。ただし、qtree ボリュームでは、スナップショットから別の名前空間または元の名前空間への復元はサポートされていません。
- 詳細設定を使用して復元操作をカスタマイズできます。詳細については、 "高度なTrident Protect復元設定を使用する"。

バックアップから別の名前空間に復元する

BackupRestore CR を使用してバックアップを別の名前空間に復元すると、 Trident Protect はアプリケーションを新しい名前空間に復元し、復元されたアプリケーションのアプリケーション CR を作成します。復元されたアプリケーションを保護するには、オンデマンド バックアップまたはスナップショットを作成するか、保護スケジュールを確立します。

既存のリソースを含む別の名前空間にバックアップを復元しても、バックアップ内のリソースと名前を共有するリソースは変更されません。バックアップ内のすべてのリソースを復元するには、ターゲット名前空間を削除して再作成するか、バックアップを新しい名前空間に復元します。

開始する前に

AWS セッション トークンの有効期限が、長時間実行される S3 復元操作に十分であることを確認します。復元操作中にトークンの有効期限が切れると、操作が失敗する可能性があります。

- 参照 "AWS API ドキュメント"現在のセッション トークンの有効期限を確認する方法の詳細については、こちらをご覧ください。
- 参照 "AWS IAM ドキュメント"AWS リソースの認証情報の詳細については、こちらをご覧ください。

Kopia をデータ ムーバーとして使用してバックアップを復元する場合、オプションで CR に注釈を指定するか、CLI を使用して Kopia が使用する一時ストレージの動作を制御できます。参照 "Kopiaのドキュメント"設定できるオプションの詳細については、こちらをご覧ください。使用 `tridentctl-protect create --help`Trident Protect CLI で注釈を指定する方法の詳細については、コマンドを参照してください。

CRを使用する

手順

1. カスタムリソース（CR）ファイルを作成し、名前を付けます `trident-protect-backup-restore-cr.yaml`。
2. 作成したファイルで、次の属性を構成します。
   - **metadata.name**: (必須) このカスタム リソースの名前。環境に合わせて一意かつ適切な名前を選択してください。
   - **spec.appArchivePath**: バックアップ コンテンツが保存される AppVault 内のパス。このパスを見つけるには、次のコマンドを使用できます。

     ```
     kubectl get backups <BACKUP_NAME> -n my-app-namespace -o
     jsonpath='{.status.appArchivePath}'
     ```

   - **spec.appVaultRef**: (必須) バックアップ コンテンツが保存される AppVault の名前。
   - **spec.namespaceMapping**: 復元操作のソース名前空間から宛先名前空間へのマッピング。交換する `my-source-namespace`そして `my-destination-namespace`あなたの環境からの情報を活用します。

     ```
     ---
     apiVersion: protect.trident.netapp.io/v1
     kind: BackupRestore
     metadata:
       name: my-cr-name
       namespace: my-destination-namespace
     spec:
       appArchivePath: my-backup-path
       appVaultRef: appvault-name
       namespaceMapping: [{"source": "my-source-namespace",
     "destination": "my-destination-namespace"}]
     ```

3. (オプション) 復元するアプリケーションの特定のリソースのみを選択する必要がある場合は、特定のラベルでマークされたリソースを含めるか除外するフィルタリングを追加します。

   Trident Protect は、選択したリソースとの関係に基づいて、いくつかのリソースを自動的に選択します。たとえば、永続ボリューム要求リソースを選択し、それに関連付けられたポッドがある場合、 Trident Protect は関連付けられたポッドも復元します。

   - **resourceFilter.resourceSelectionCriteria**: (フィルタリングに必須) 使用 `Include`または `Exclude`resourceMatchers で定義されたリソースを含めるか除外するかを指定します。含めるまたは除外するリソースを定義するには、次の resourceMatchers パラメータを追加します。
     - **resourceFilter.resourceMatchers**: resourceMatcher オブジェクトの配列。この配列に複数の要素を定義すると、それらは OR 演算として一致し、各要素内のフィールド (グループ、種類、バージョン) は AND 演算として一致します。

- **resourceMatchers[].group**: (オプション) フィルタリングするリソースのグループ。
- **resourceMatchers[].kind**: (オプション) フィルタリングするリソースの種類。
- **resourceMatchers[].version**: (オプション) フィルタリングするリソースのバージョン。
- **resourceMatchers[].names**: (オプション) フィルタリングするリソースのKubernetes metadata.name フィールド内の名前。
- **resourceMatchers[].namespaces**: (オプション) フィルタリングするリソースのKubernetes metadata.name フィールド内の名前空間。
- **resourceMatchers[].labelSelectors**: (オプション) Kubernetesのmetadata.nameフィールドのラベルセレクタ文字列。 "Kubernetesドキュメント" 。例えば： `"trident.netapp.io/os=linux"` 。

  例えば：

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. 入力したら `trident-protect-backup-restore-cr.yaml`正しい値を持つファイルには、CR を適用します。

```
kubectl apply -f trident-protect-backup-restore-cr.yaml
```

CLIを使用する

手順

1. 括弧内の値を環境の情報に置き換えて、バックアップを別の名前空間に復元します。その `namespace-mapping`引数はコロンで区切られた名前空間を使用して、ソース名前空間を正しい宛先名前空間にマッピングします。 `source1:dest1,source2:dest2` 。例えば：

```
tridentctl-protect create backuprestore <my_restore_name> \
--backup <backup_namespace>/<backup_to_restore> \
--namespace-mapping <source_to_destination_namespace_mapping> \
-n <application_namespace>
```

バックアップから元の名前空間に復元する

いつでもバックアップを元の名前空間に復元できます。

開始する前に

AWS セッション トークンの有効期限が、長時間実行される S3 復元操作に十分であることを確認します。復元操作中にトークンの有効期限が切れると、操作が失敗する可能性があります。

- 参照 "AWS API ドキュメント"現在のセッション トークンの有効期限を確認する方法の詳細については、こちらをご覧ください。
- 参照 "AWS IAM ドキュメント"AWS リソースの認証情報の詳細については、こちらをご覧ください。

> Kopia をデータ ムーバーとして使用してバックアップを復元する場合、オプションで CR に注釈を指定するか、CLI を使用して Kopia が使用する一時ストレージの動作を制御できます。参照 "Kopiaのドキュメント"設定できるオプションの詳細については、こちらをご覧ください。使用 `tridentctl-protect create --help`Trident Protect CLI で注釈を指定する方法の詳細については、コマンドを参照してください。

CRを使用する

手順

1. カスタムリソース（CR）ファイルを作成し、名前を付けます `trident-protect-backup-ipr-cr.yaml`。
2. 作成したファイルで、次の属性を構成します。
   - **metadata.name**: (必須) このカスタム リソースの名前。環境に合わせて一意かつ適切な名前を選択してください。
   - **spec.appArchivePath**: バックアップ コンテンツが保存される AppVault 内のパス。このパスを見つけるには、次のコマンドを使用できます。

     ```
     kubectl get backups <BACKUP_NAME> -n my-app-namespace -o
     jsonpath='{.status.appArchivePath}'
     ```

   - **spec.appVaultRef**: (必須) バックアップ コンテンツが保存される AppVault の名前。

     例えば：

```
---
apiVersion: protect.trident.netapp.io/v1
kind: BackupInplaceRestore
metadata:
  name: my-cr-name
  namespace: my-app-namespace
spec:
  appArchivePath: my-backup-path
  appVaultRef: appvault-name
```

3. (オプション) 復元するアプリケーションの特定のリソースのみを選択する必要がある場合は、特定のラベルでマークされたリソースを含めるか除外するフィルタリングを追加します。

   Trident Protect は、選択したリソースとの関係に基づいて、いくつかのリソースを自動的に選択します。たとえば、永続ボリューム要求リソースを選択し、それに関連付けられたポッドがある場合、 Trident Protect は関連付けられたポッドも復元します。

   - **resourceFilter.resourceSelectionCriteria**: (フィルタリングに必須) 使用 `Include`または `Exclude`resourceMatchers で定義されたリソースを含めるか除外するかを指定します。含めるまたは除外するリソースを定義するには、次の resourceMatchers パラメータを追加します。
     - **resourceFilter.resourceMatchers**: resourceMatcher オブジェクトの配列。この配列に複数の要素を定義すると、それらは OR 演算として一致し、各要素内のフィールド (グループ、種類、バージョン) は AND 演算として一致します。
       - **resourceMatchers[].group**: (オプション) フィルタリングするリソースのグループ。
       - **resourceMatchers[].kind**: (オプション) フィルタリングするリソースの種類。

- **resourceMatchers[].version**: (オプション) フィルタリングするリソースのバージョン。
- **resourceMatchers[].names**: (オプション) フィルタリングするリソースのKubernetes metadata.name フィールド内の名前。
- **resourceMatchers[].namespaces**: (オプション) フィルタリングするリソースのKubernetes metadata.name フィールド内の名前空間。
- **resourceMatchers[].labelSelectors**: (オプション) Kubernetesのmetadata.nameフィールドのラベルセレクタ文字列。 "Kubernetesドキュメント" 。例えば： `"trident.netapp.io/os=linux"` 。

  例えば：

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. 入力したら `trident-protect-backup-ipr-cr.yaml`正しい値を持つファイルには、CR を適用します。

```
kubectl apply -f trident-protect-backup-ipr-cr.yaml
```

CLIを使用する

手順

1. 括弧内の値を環境の情報に置き換えて、バックアップを元の名前空間に復元します。その `backup` 引数は、名前空間とバックアップ名を次の形式で使用します。 `<namespace>/<name>` 。例えば：

```
tridentctl-protect create backupinplacerestore <my_restore_name> \
--backup <namespace/backup_to_restore> \
-n <application_namespace>
```

バックアップから別のクラスターに復元する

元のクラスターに問題がある場合は、バックアップを別のクラスターに復元できます。

> Kopia をデータ ムーバーとして使用してバックアップを復元する場合、オプションで CR に注釈を指定するか、CLI を使用して Kopia が使用する一時ストレージの動作を制御できます。参照 "Kopiaのドキュメント"設定できるオプションの詳細については、こちらをご覧ください。使用 `tridentctl-protect create --help`Trident Protect CLI で注釈を指定する方法の詳細については、コマンドを参照してください。

開始する前に

次の前提条件が満たされていることを確認してください。

- 宛先クラスターにはTrident Protect がインストールされています。
- 宛先クラスターは、バックアップが保存されているソース クラスターと同じ AppVault のバケット パスにアクセスできます。
- 実行時に、ローカル環境がAppVault CRで定義されたオブジェクトストレージバケットに接続できることを確認してください。 `tridentctl-protect get appvaultcontent`指示。ネットワーク制限によりアクセスできない場合は、代わりに宛先クラスターのポッド内からTrident Protect CLI を実行します。
- AWS セッショントークンの有効期限が、長時間実行される復元操作に十分であることを確認します。復元操作中にトークンの有効期限が切れると、操作が失敗する可能性があります。
  - 参照 "AWS API ドキュメント"現在のセッション トークンの有効期限を確認する方法の詳細については、こちらをご覧ください。
  - 参照 "AWSのドキュメント"AWS リソースの認証情報の詳細については、こちらをご覧ください。

手順

1. Trident Protect CLI プラグインを使用して、宛先クラスター上の AppVault CR の可用性を確認します。

```
tridentctl-protect get appvault --context <destination_cluster_name>
```

> アプリケーションの復元を目的とした名前空間が宛先クラスターに存在することを確認します。

2. 宛先クラスターから利用可能な AppVault のバックアップ コンテンツを表示します。

```
tridentctl-protect get appvaultcontent <appvault_name> \
--show-resources backup \
--show-paths \
--context <destination_cluster_name>
```

このコマンドを実行すると、AppVault 内の使用可能なバックアップ（元のクラスター、対応するアプリケーション名、タイムスタンプ、アーカイブ パスなど）が表示されます。

出力例:

```
+-------------+-----------+--------+-----------------
+--------------------------+-------------+
|   CLUSTER   |    APP    |  TYPE  |      NAME       |        TIMESTAMP
|    PATH     |
+-------------+-----------+--------+-----------------
+--------------------------+-------------+
| production1 | wordpress | backup | wordpress-bkup-1| 2024-10-30
08:37:40 (UTC)| backuppath1 |
| production1 | wordpress | backup | wordpress-bkup-2| 2024-10-30
08:37:40 (UTC)| backuppath2 |
+-------------+-----------+--------+-----------------
+--------------------------+-------------+
```

3. AppVault 名とアーカイブ パスを使用して、アプリケーションを宛先クラスターに復元します。

CRを使用する

1. カスタムリソース（CR）ファイルを作成し、名前を付けます `trident-protect-backup-restore-cr.yaml`。
2. 作成したファイルで、次の属性を構成します。
   - **metadata.name**: (必須) このカスタム リソースの名前。環境に合わせて一意かつ適切な名前を選択してください。
   - **spec.appVaultRef**: (必須) バックアップ コンテンツが保存される AppVault の名前。
   - **spec.appArchivePath**: バックアップ コンテンツが保存される AppVault 内のパス。このパスを見つけるには、次のコマンドを使用できます。

     ```
     kubectl get backups <BACKUP_NAME> -n my-app-namespace -o
     jsonpath='{.status.appArchivePath}'
     ```

     BackupRestore CR が使用できない場合は、手順 2 に記載されているコマンドを使用してバックアップの内容を表示できます。

   - **spec.namespaceMapping**: 復元操作のソース名前空間から宛先名前空間へのマッピング。交換する `my-source-namespace`そして `my-destination-namespace`あなたの環境からの情報を活用します。

     例えば：

```
apiVersion: protect.trident.netapp.io/v1
kind: BackupRestore
metadata:
  name: my-cr-name
  namespace: my-destination-namespace
spec:
  appVaultRef: appvault-name
  appArchivePath: my-backup-path
  namespaceMapping: [{"source": "my-source-namespace", "
destination": "my-destination-namespace"}]
```

3. 入力したら `trident-protect-backup-restore-cr.yaml`正しい値を持つファイルには、CR を適用します。

```
kubectl apply -f trident-protect-backup-restore-cr.yaml
```

CLIを使用する

1. 次のコマンドを使用してアプリケーションを復元し、括弧内の値を環境の情報に置き換えます。namespace-mapping 引数は、コロンで区切られた名前空間を使用して、source1:dest1

、source2:dest2 の形式でソース名前空間を正しい宛先名前空間にマッピングします。例えば：

```
tridentctl-protect create backuprestore <restore_name> \
--namespace-mapping <source_to_destination_namespace_mapping> \
--appvault <appvault_name> \
--path <backup_path> \
--context <destination_cluster_name> \
-n <application_namespace>
```

スナップショットから別の名前空間に復元する

カスタム リソース (CR) ファイルを使用して、スナップショットからデータを別の名前空間または元のソース名前空間に復元できます。SnapshotRestore CR を使用してスナップショットを別の名前空間に復元すると、Trident Protect はアプリケーションを新しい名前空間に復元し、復元されたアプリケーションのアプリケーション CR を作成します。復元されたアプリケーションを保護するには、オンデマンド バックアップまたはスナップショットを作成するか、保護スケジュールを確立します。

SnapshotRestoreは、 `spec.storageClassMapping`属性ですが、ソース ストレージ クラスと宛先ストレージ クラスが同じストレージ バックエンドを使用する場合のみです。復元しようとすると、 `StorageClass`異なるストレージ バックエンドを使用する場合、復元操作は失敗します。

開始する前に

AWS セッション トークンの有効期限が、長時間実行される S3 復元操作に十分であることを確認します。復元操作中にトークンの有効期限が切れると、操作が失敗する可能性があります。

- 参照 "AWS API ドキュメント"現在のセッション トークンの有効期限を確認する方法の詳細については、こちらをご覧ください。
- 参照 "AWS IAM ドキュメント"AWS リソースの認証情報の詳細については、こちらをご覧ください。

CRを使用する

手順

1. カスタムリソース（CR）ファイルを作成し、名前を付けます `trident-protect-snapshot-restore-cr.yaml`。
2. 作成したファイルで、次の属性を構成します。
   - **metadata.name**: (必須) このカスタム リソースの名前。環境に合わせて一意かつ適切な名前を選択してください。
   - **spec.appVaultRef**: (必須) スナップショットの内容が保存される AppVault の名前。
   - **spec.appArchivePath**: スナップショットの内容が保存される AppVault 内のパス。このパスを見つけるには、次のコマンドを使用できます。

     ```
     kubectl get snapshots <SNAPHOT_NAME> -n my-app-namespace -o
     jsonpath='{.status.appArchivePath}'
     ```

   - **spec.namespaceMapping**: 復元操作のソース名前空間から宛先名前空間へのマッピング。交換する `my-source-namespace`そして `my-destination-namespace`あなたの環境からの情報を活用します。

     ```
     ---
     apiVersion: protect.trident.netapp.io/v1
     kind: SnapshotRestore
     metadata:
       name: my-cr-name
       namespace: my-app-namespace
     spec:
       appVaultRef: appvault-name
       appArchivePath: my-snapshot-path
       namespaceMapping: [{"source": "my-source-namespace",
     "destination": "my-destination-namespace"}]
     ```

3. (オプション) 復元するアプリケーションの特定のリソースのみを選択する必要がある場合は、特定のラベルでマークされたリソースを含めるか除外するフィルタリングを追加します。

   Trident Protect は、選択したリソースとの関係に基づいて、いくつかのリソースを自動的に選択します。たとえば、永続ボリューム要求リソースを選択し、それに関連付けられたポッドがある場合、 Trident Protect は関連付けられたポッドも復元します。

   - **resourceFilter.resourceSelectionCriteria**: (フィルタリングに必須) 使用 `Include`または `Exclude`resourceMatchers で定義されたリソースを含めるか除外するかを指定します。含めるまたは除外するリソースを定義するには、次の resourceMatchers パラメータを追加します。
     - **resourceFilter.resourceMatchers**: resourceMatcher オブジェクトの配列。この配列に複数の要素を定義すると、それらは OR 演算として一致し、各要素内のフィールド (グループ、種類、バージョン) は AND 演算として一致します。

- **resourceMatchers[].group**: (オプション) フィルタリングするリソースのグループ。
- **resourceMatchers[].kind**: (オプション) フィルタリングするリソースの種類。
- **resourceMatchers[].version**: (オプション) フィルタリングするリソースのバージョン。
- **resourceMatchers[].names**: (オプション) フィルタリングするリソースのKubernetes metadata.name フィールド内の名前。
- **resourceMatchers[].namespaces**: (オプション) フィルタリングするリソースのKubernetes metadata.name フィールド内の名前空間。
- **resourceMatchers[].labelSelectors**: (オプション) Kubernetesのmetadata.nameフィールドのラベルセレクタ文字列。 "Kubernetesドキュメント" 。例えば： `"trident.netapp.io/os=linux"` 。

  例えば：

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. 入力したら `trident-protect-snapshot-restore-cr.yaml`正しい値を持つファイルには、CR を適用します。

```
kubectl apply -f trident-protect-snapshot-restore-cr.yaml
```

CLIを使用する

手順

1. 括弧内の値を環境の情報に置き換えて、スナップショットを別の名前空間に復元します。
   - その `snapshot`引数は、名前空間とスナップショット名を次の形式で使用します。 `<namespace>/<name>` 。
   - その `namespace-mapping`引数はコロンで区切られた名前空間を使用して、ソース名前空間を正しい宛先名前空間にマッピングします。 `source1:dest1,source2:dest2` 。

例えば：

```
tridentctl-protect create snapshotrestore <my_restore_name> \
--snapshot <namespace/snapshot_to_restore> \
--namespace-mapping <source_to_destination_namespace_mapping> \
-n <application_namespace>
```

スナップショットから元の名前空間に復元する

いつでもスナップショットを元の名前空間に復元できます。

> アプリケーションが複数の名前空間を使用し、これらの名前空間に同じ名前のPVCがある場合、スナップショットの復元操作（インプレースおよび新しい名前空間への復元の両方）は正しく機能しません。復元されたすべてのボリュームには、名前空間ごとに正しいデータではなく、同じデータが含まれます。スナップショットの復元ではなくバックアップの復元を使用するか、この問題を修正するバージョン26.02以降にアップグレードしてください。

開始する前に

AWS セッション トークンの有効期限が、長時間実行される S3 復元操作に十分であることを確認します。復元操作中にトークンの有効期限が切れると、操作が失敗する可能性があります。

- 参照 "AWS API ドキュメント"現在のセッション トークンの有効期限を確認する方法の詳細については、こちらをご覧ください。
- 参照 "AWS IAM ドキュメント"AWS リソースの認証情報の詳細については、こちらをご覧ください。

CRを使用する

手順

1. カスタムリソース（CR）ファイルを作成し、名前を付けます `trident-protect-snapshot-ipr-cr.yaml`。
2. 作成したファイルで、次の属性を構成します。
   - **metadata.name**: (必須) このカスタム リソースの名前。環境に合わせて一意かつ適切な名前を選択してください。
   - **spec.appVaultRef**: (必須) スナップショットの内容が保存される AppVault の名前。
   - **spec.appArchivePath**: スナップショットの内容が保存される AppVault 内のパス。このパスを見つけるには、次のコマンドを使用できます。

```
kubectl get snapshots <SNAPSHOT_NAME> -n my-app-namespace -o
jsonpath='{.status.appArchivePath}'
```

```
---
apiVersion: protect.trident.netapp.io/v1
kind: SnapshotInplaceRestore
metadata:
  name: my-cr-name
  namespace: my-app-namespace
spec:
  appVaultRef: appvault-name
    appArchivePath: my-snapshot-path
```

3. (オプション) 復元するアプリケーションの特定のリソースのみを選択する必要がある場合は、特定のラベルでマークされたリソースを含めるか除外するフィルタリングを追加します。

> Trident Protect は、選択したリソースとの関係に基づいて、いくつかのリソースを自動的に選択します。たとえば、永続ボリューム要求リソースを選択し、それに関連付けられたポッドがある場合、 Trident Protect は関連付けられたポッドも復元します。

   - **resourceFilter.resourceSelectionCriteria**: (フィルタリングに必須) 使用 `Include`または `Exclude`resourceMatchers で定義されたリソースを含めるか除外するかを指定します。含めるまたは除外するリソースを定義するには、次の resourceMatchers パラメータを追加します。
     - **resourceFilter.resourceMatchers**: resourceMatcher オブジェクトの配列。この配列に複数の要素を定義すると、それらは OR 演算として一致し、各要素内のフィールド (グループ、種類、バージョン) は AND 演算として一致します。
       - **resourceMatchers[].group**: (オプション) フィルタリングするリソースのグループ。
       - **resourceMatchers[].kind**: (オプション) フィルタリングするリソースの種類。
       - **resourceMatchers[].version**: (オプション) フィルタリングするリソースのバージョン。
       - **resourceMatchers[].names**: (オプション) フィルタリングするリソースのKubernetes

metadata.name フィールド内の名前。

- **resourceMatchers[].namespaces**: (オプション) フィルタリングするリソースのKubernetes metadata.name フィールド内の名前空間。
- **resourceMatchers[].labelSelectors**: (オプション) Kubernetesのmetadata.nameフィールドのラベルセレクタ文字列。 "Kubernetesドキュメント" 。例えば： `"trident.netapp.io/os=linux"` 。

  例えば：

```
spec:
  resourceFilter:
    resourceSelectionCriteria: "Include"
    resourceMatchers:
      - group: my-resource-group-1
        kind: my-resource-kind-1
        version: my-resource-version-1
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
      - group: my-resource-group-2
        kind: my-resource-kind-2
        version: my-resource-version-2
        names: ["my-resource-names"]
        namespaces: ["my-resource-namespaces"]
        labelSelectors: ["trident.netapp.io/os=linux"]
```

4. 入力したら `trident-protect-snapshot-ipr-cr.yaml`正しい値を持つファイルには、CR を適用します。

```
kubectl apply -f trident-protect-snapshot-ipr-cr.yaml
```

CLIを使用する

手順

1. 括弧内の値を環境の情報に置き換えて、スナップショットを元の名前空間に復元します。例えば：

```
tridentctl-protect create snapshotinplacerestore <my_restore_name> \
--snapshot <snapshot_to_restore> \
-n <application_namespace>
```

復元操作のステータスを確認する

コマンド ラインを使用して、進行中、完了、または失敗した復元操作のステータスを確認できます。

手順

1. 復元操作のステータスを取得するには、次のコマンドを使用します。括弧内の値は、ご使用の環境の情報に置き換えてください。

```
kubectl get backuprestore -n <namespace_name> <my_restore_cr_name> -o
jsonpath='{.status}'
```

高度なTrident Protect復元設定を使用する

注釈、名前空間設定、ストレージ オプションなどの詳細設定を使用して、特定の要件を満たすように復元操作をカスタマイズできます。

復元およびフェイルオーバー操作中の名前空間の注釈とラベル

復元およびフェイルオーバー操作中に、宛先名前空間のラベルと注釈は、ソース名前空間のラベルと注釈と一致するように作成されます。宛先名前空間に存在しないソース名前空間のラベルまたは注釈が追加され、既存のラベルまたは注釈はソース名前空間の値と一致するように上書きされます。宛先名前空間にのみ存在するラベルまたは注釈は変更されません。

Red Hat OpenShift を使用する場合は、OpenShift 環境における名前空間アノテーションの重要な役割に注意することが重要です。名前空間アノテーションにより、復元されたポッドが OpenShift のセキュリティ コンテキスト制約 (SCC) によって定義された適切な権限とセキュリティ構成に準拠し、権限の問題なくボリュームにアクセスできるようになります。詳細については、 "OpenShift セキュリティ コンテキスト制約のドキュメント" 。

Kubernetes環境変数を設定することで、宛先名前空間内の特定のアノテーションが上書きされるのを防ぐことができます。 `RESTORE_SKIP_NAMESPACE_ANNOTATIONS`復元またはフェイルオーバー操作を実行する前に。例えば：

```
helm upgrade trident-protect --set
restoreSkipNamespaceAnnotations=<annotation_key_to_skip_1>,<annotation_key
_to_skip_2> --reuse-values
```

復元またはフェイルオーバー操作を実行する場合、 `restoreSkipNamespaceAnnotations` そして `restoreSkipNamespaceLabels` 復元またはフェイルオーバー操作から除外されます。これらの設定が Helm の初期インストール時に構成されていることを確認してください。詳細については、 "AutoSupportと名前空間フィルタリングオプションを構成する"。

ソースアプリケーションをHelmを使用してインストールした場合、 `--create-namespace`旗には特別な扱いが与えられます `name`ラベルキー。復元またはフェイルオーバー プロセス中に、 Trident Protect はこのラベルを宛先名前空間にコピーしますが、ソースからの値がソース名前空間と一致する場合は、値を宛先名前空間の値に更新します。この値がソース名前空間と一致しない場合は、変更されずに宛先名前空間にコピーされます。

例

次の例は、それぞれ異なる注釈とラベルを持つソース名前空間と宛先名前空間を示しています。操作の前後の宛先名前空間の状態や、宛先名前空間で注釈とラベルがどのように結合または上書きされるかを確認できます。

復元またはフェイルオーバー操作の前に

次の表は、復元またはフェイルオーバー操作前のサンプルのソース名前空間と宛先名前空間の状態を示しています。

ネームスペース	アノテーション	ラベル
名前空間 ns-1 (ソース)	• アノテーション.one/キー: "更新された値" • アノテーション.2/キー: "true"	• 環境=本番環境 • コンプライアンス=HIPAA • 名前=ns-1
名前空間 ns-2 (宛先)	• アノテーション.one/キー: "true" • 注釈.three/キー: "false"	• 役割=データベース

復元操作後

次の表は、復元またはフェイルオーバー操作後の宛先名前空間の例の状態を示しています。いくつかのキーが追加され、いくつかは上書きされ、 `name`ラベルは宛先名前空間と一致するように更新されました:

ネームスペース	アノテーション	ラベル
名前空間 ns-2 (宛先)	• アノテーション.one/キー: "更新された値" • アノテーション.2/キー: "true" • 注釈.three/キー: "false"	• 名前=ns-2 • コンプライアンス=HIPAA • 環境=本番環境 • 役割=データベース

サポートされているフィールド

このセクションでは、復元操作に使用できる追加のフィールドについて説明します。

ストレージクラスのマッピング

その `spec.storageClassMapping`属性は、ソース アプリケーションに存在するストレージ クラスからターゲット クラスター上の新しいストレージ クラスへのマッピングを定義します。これは、異なるストレージ クラスを持つクラスター間でアプリケーションを移行する場合や、BackupRestore 操作のストレージ バックエンドを変更する場合に使用できます。

例：

```
storageClassMapping:
  - destination: "destinationStorageClass1"
    source: "sourceStorageClass1"
  - destination: "destinationStorageClass2"
    source: "sourceStorageClass2"
```

サポートされている注釈

このセクションでは、システム内のさまざまな動作を構成するためにサポートされているアノテーションを一覧表示します。ユーザーが注釈を明示的に設定しない場合、システムはデフォルト値を使用します。

注釈	タイプ	説明	デフォルト値
protect.trident.netapp.io/データムーバータイムアウト秒	string	データ ムーバー操作を停止できる最大時間 (秒単位)。	「300」
protect.trident.netapp.io/kopia-content-cache-size-limit-mb	string	Kopia コンテンツ キャッシュの最大サイズ制限 (メガバイト単位)。	「1000」

著作権に関する情報

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータ ソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用権を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用権については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

商標に関する情報

NetApp、NetAppのロゴ、http://www.netapp.com/TMに記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。